

広島大学学術情報リポジトリ

Hiroshima University Institutional Repository

Title	増加の意味解釈とユニフィケーション
Auther(s)	林, 勲
Citation	ニダバ , 20 : 11 - 18
Issue Date	1991-03-31
DOI	
Self DOI	
URL	http://ir.lib.hiroshima-u.ac.jp/00047215
Right	
Relation	



増加的意味解釈とユニフィケーション

林 勲

0. はじめに

この論文で論議される文法は、ユニフィケーション操作を基底とする自然言語処理に動機づけられている。それはカテゴリーアル文法のより柔軟な拡張の試みに属している。その文法を、ここでは仮にユニフィケーション基底結合カテゴリーアル文法 (Unification-based Combinatory Categorical Grammar (UCCG)) と呼ぶことにする。何故なら、その文法が Ades & Steedman (1982) によって提案されそして Steedman (1985, 1987, 1989) によって展開されている結合カテゴリーアル文法 (Combinatory Categorical Grammar (CCG)) と、Zeevat (1988) によるユニフィケーションカテゴリーアル文法 (Unification Categorical Grammar (UCG)) のいずれにも類似した、いわば折衷的なものだからである。すなわち、UCCGは、CCGにおけるように増加的意味解釈を仮定し、結合ルールとして関数合成に大きく依存していると同時に、UCGにおけるようにユニフィケーション操作を基底とした文法記述を行なうからである。

しかしながら、UCCGには重要な点において両者との相違が存在する。それはユニフィケーション操作に基づいて関数適用のみならず関数合成をも説明する点である。Steedmanはユニフィケーションの概念を用いてはいるが、その使用に関数適用の意味論的説明のためだけに限定している。一方、Zeevatはユニフィケーション操作に基本的に依存しているが、結合ルールとして関数適用のみを用い、関数合成は用いていない。

この論文の目的は、UCCGの概略を素描することと、ユニフィケーションの概念を用いて関数合成を説明することである。そうすることによって、増加的意味解釈の仮定に基づくユニフィケーション基底のカテゴリーアルシステムへの手掛かりを求めることにある。

1. 統語カテゴリーとユニフィケーション

UCCGにおける統語カテゴリーは、基本カテゴリーと関数子カテゴリーから成る。基本カテゴリーは、たとえば N (名詞) とか S (文) といったものである。関数子カテゴリーは、たとえば $S \backslash NP$ (自動詞) とか VP / NP (他動詞) とか $S / (S \backslash NP)$ (名詞句、NPとも略される) などのように、スラッシュを含む表記が与えられるものである。 X/Y , $(W/X)/Y$ のような形式をもつ関数子カテゴリーは、その右側に存在する Y にたいする関数と解釈され、それぞれカテゴ

リー Y からカテゴリー X への関数、カテゴリー Y からカテゴリー W/X への関数、を指示する。また、 $X \setminus Y$, $(W/X) \setminus Y$ のような形式をもつ関数子カテゴリーは、その左側に存在する Y にたいする関数と解釈され、それぞれカテゴリー Y からカテゴリー X への関数、カテゴリー Y からカテゴリー W/X への関数、を指示する。

関数子の方向性（つまり、スラッシュ表記）については、UCCGはCCGの表記法を採用している。すなわち、スラッシュの左側には常に結果のカテゴリーが、スラッシュの右側には常に関数の項カテゴリーが表示される。スラッシュ $/$ は、その関数子の項が関数子の右側に存在することを、またスラッシュ \backslash は、その関数子の項が関数子の左側に存在することを意味している。この表記法によって、複合統語カテゴリーにおける結果カテゴリーと項カテゴリーの位置関係について、統一的な表記が与えられる。

ところで、UCCGの関数適用および関数合成においては、関数子の活動的部分の概念がユニフィケーション操作にとって重要な役割を果たす。関数子の活動的部分とは、ユニフィケーション操作が実際に行なわれる場を意味している。関数適用における関数子の活動的部分と関数合成における関数子の活動的部分は、それぞれ (1) と (2) によって与えられるようなものである。

- (1) 関数適用においては、関数子カテゴリー A/B または $A \setminus B$ における B を、それぞれの関数子カテゴリーの活動的部分と呼ぶ。
- (2) 関数合成においては、合成される二つの関数子カテゴリー A/B , B/C におけるそれぞれの B を、それぞれの関数子カテゴリーの活動的部分と呼ぶ。

UCCGにおけるユニフィケーションの概念は、関数適用および関数合成において、単に統語論的にだけでなく意味論的にも、基本的に重要な役割を果たす。何故なら、ユニフィケーション操作は、統語情報および意味情報をともに含む単一のデータ構造によって演算可能な形式を与えられるカテゴリー表示（後に述べる）にたいして作用するからである。

よく知られているように、二つの表示のユニフィケーションは、もしその情報が互いに矛盾しないならば、最初の二つのすべての情報を結合するような第三の表示である。すなわち、不完全な部分的情報が互いに融合され（情報はいずれの方向にも流れることができる）、より完全な明細化を与えるような操作である。もしその情報が矛盾するなら、ユニフィケーションは失敗する。

UCCGにおける関数適用および関数合成は、ともにユニフィケーション操作に基づいて、次の三段階を経ることによってなされる。

- (i) インスタンスエイション：関数適用においては、関数子カテゴリー表示の活動的部分と項カテゴリー表示とを、関数合成においては、二つの関数子カテゴリー表示の活動的部分をユニファイする。これによって、それぞれの活動的部分のより完全なカテゴリー表示を得る。
- (ii) リエントランス：インスタンスエイションの副次的効果として、ユニフィケーション

によって得られた新しい情報の一部を結果カテゴリー表示の対応する部分へと受け渡す。これによって、結果カテゴリー表示がより完全に明細化される。

- (iii) ストリッピング：インスタンス化およびリエントランスを終了した関数カテゴリー表示の活動的部分を削除する。これによって、より完全に明細化された結果カテゴリー表示が得られ、二つの表現の結合が完了する。

UCCGにおいて注目すべき重要なことは、ユニフィケーション操作が関数適用のみならず関数合成にたいしても、同様に基本的な操作として用いられることである。Steedmanのようにユニフィケーションを関数適用のみに限定すべき理由を、私は見いだせない。何故なら、このようなユニフィケーションの限定的使用は、理論の中にad hoc性を持ち込むことになり、それによって理論の首尾一貫性が損なわれるからである。また、Zeevatのように関数合成を理論から排除しなければならない理由も見いだせない。何故なら、もし増加の意味解釈を仮定するならば、関数合成はそれを可能にする基本的結合ルールだからである。

3. カテゴリー表示

UCCGの簡略化されたカテゴリー表示は、(i) 統語タイプCと(ii) 意味表示srの組み合わせから成る複合表示が与えられる。たとえば、Aの形式をもつ単純な統語カテゴリーのカテゴリー表示は、 $C_i:sr_i$ である。また、A/Bまたは $A \setminus B$ の形式をもつ複合統語カテゴリーのカテゴリー表示は、それぞれ $C_i:sr_i/C_j:sr_j$ または $C_i:sr_i \setminus C_j:sr_j$ である。このように複合統語カテゴリーのカテゴリー表示は、スラッシュで分離された部分に個々の単純な統語カテゴリーのカテゴリー表示が与えられる。意味表示srは、述語論理の言語への翻訳によって小文字で表記されるが、それらは定項（'を付して表記される）または変項、またはそれらの組み合わせから成っている。

具体例として、たとえば自動詞walksのカテゴリー表示は、次の(3)によって与えられるだろう。

(3) walks

$S:walk' \ npl \setminus NP:npl$

このカテゴリー表示が示していることは次のようなことである。つまり、統語タイプがNPであり意味表示がnplであるようなカテゴリー表示をもつ、左方に存在する表現と関数適用によって結合し、結果として、統語タイプがSであり意味表示が $walk' \ npl$ であるような表現へと結果するような要素であることを意味している。

また、もしこの自動詞walksの左方に存在する表現が名詞句Harryであるとすれば、そのカテゴリー表示は、次の(4)によって与えられるだろう。

(4) Harry

$NP:harry'$

今や、前のセクションで述べた三段階の手順にしたがって、関数適用が行なわれる準備が

できている。(3)におけるNP:nplはカテゴリー表示の活動的部分であり、(4)のNP:nplはその活動的部分とユニファイされる項カテゴリー表示である。ユニフィケーション操作に基づく関数適用の結果は、次の(5)のようになるだろう。

(5) Harry walks
S:walk' harry'

しかしながら、(4)のように与えられた名詞句のカテゴリー表示は、増加の意味解釈に基づく実際のUCCGにおいては、しばしば多少異なった表示が与えられる。Montague (1974)にしたがって、次の(6)のようなタイプ繰り上げされた名詞句としてのカテゴリー表示がそれである。

(6) Harry
S:pred harry' / (S:pred harry' \NP:harry')

このようにタイプ繰り上げされた名詞句(6)は、今や(3)を項カテゴリーとして関数適用できるような関数カテゴリーとなっている。したがって、Harry walksのような連鎖の増加の意味解釈の派生の全過程を、再度簡略化することなく表示するならば、次の(7)のようになるだろう。

(7) Harry [Cat, R] walks [Cat, R]
i) NP:harry' [T, R] S:walk' npl \NP:npl
ii) S:pred harry' / (S:pred harry' \NP:harry')
iii) S:pred harry' / (S:walk' harry' \NP:harry') [Re.]
iv) S:walk' harry' / (S:walk' harry' \NP:harry') [St.]
v) S:walk' harry'

(7i)は、主語名詞句Harryおよび述語動詞walksのそれぞれのカテゴリー表示 [Cat, R] である。次に(6)に与えられているように、名詞句はタイプ繰り上げ [T, R] され (7ii) によって示される関数子カテゴリーとなる。この関数子カテゴリー表示の (S:pred harry' \NP:harry') で表されている部分は、関数子カテゴリー表示の活動的部分である。かくして、関数適用が行なわれる。まず、この活動的部分と項カテゴリーの S:walk' npl \NP:npl がユニファイされることにより、インスタンスエイション [In.] が行なわれ (7iii) が得られる。次にリエントランス [Re.] 効果は、結果カテゴリー表示の意味表示中の pred を walk' へと明細化して (7iv) の表示を与える。関数適用の最終段階は、ストリップング [St.] によりカテゴリー表示の活動的部分を削除して (7v) を与える。このようにして、連鎖 Harry walks にたいして、統語タイプとして S (すなわち文)、意味表示として walk' harry' をもつようなカテゴリー表示を得ることができる。これがユニフィケーション操作に基づく関数適用の説明である。

4. 増加の意味解釈

UCCGは、CCGにしたがって、増加の意味解釈の仮定に基づいている。すなわち、我々が文

を理解する際に、各語に出会うにしたがって、次第に意味を集積していくという形で連続する語を順次処理していくことを前提としている。このような仮定は、標準的文法理論の放棄を要求する。その結果、伝統的アプローチでは結合不可能な種類の連鎖でさえも構成要素として取り扱うことが可能になる。また、そのような構成要素を意味解釈の単位とすることを許すような形式でシステムが構成されるだろう。さらに、通常は右方枝分かれ構造が与えられるような連鎖にたいしても、左方枝分かれ構造分析を与えることを要求するだろう。

増加の意味解釈を可能にするために、関数適用ルールに加えて、関数合成ルールやタイプ繰り上げルールなどが、結合ルールとして導入されている。特に関数合成ルールは、カテゴリーの単純な関数適用から結果するような右方枝分かれ構造を左方枝分かれ構造へと変える効果をもっている。そして、これらの結合ルールは、すべて意味解釈と直接的に不可分の緊密な関係をもっているために、UCCGは統語論と意味論の同型性を保持している。

カテゴリー構造の簡略化された表示が、統語タイプと意味解釈を合わせもつような単一のデータ構造によって、演算可能な形式で与えられることは既に述べた。このために、UCGにおいては、UCGにおけると同様に、ユニフィケーション操作が基本的役割を果たすことができる。しかも、このユニフィケーション操作は、CCGやUCGと違って、関数適用だけでなく関数合成にたいしても拡張されている。

UCCGが、ユニフィケーション操作に基づいて、関数適用と同様に関数合成も極めてエレガントに説明することができることを次に示したい。たとえば、Tom eats applesという文を例にとろう。増加の意味解釈は、この文の分析として関数合成と関数適用をともに要求する。その派生は、次の(8)に示されている。

$$\begin{array}{l}
 (8) \quad \text{Tom} \qquad \qquad \qquad \text{eats} \qquad \qquad \qquad \text{apples} \\
 \hline
 \text{S:pred tom' / (S:pred tom' \backslash NP:tom')} \quad (\text{S:eat' np2 np1 \backslash NP:np1}) / \text{NP:np2} \quad \text{NP:apples} \\
 \hline
 \text{S:eat' np2 tom' / (S:eat' np2 tom' \backslash NP:tom')} / \text{NP:np2} \\
 \hline
 \text{S:eat' np2 tom' / NP:np2} \\
 \hline
 \text{S:eat' apples' tom' / NP:apples} \\
 \hline
 \text{S:eat' apples' tom'}
 \end{array}$$

この場合、UCCGの増加の意味解釈は、先ず、Tom eatsという連鎖を構成要素とみなすだろう。ところで、名詞句Tomと他動詞eatsは、それぞれA/B、B/Cの形式で表されるような関数子カテゴリー表示をもっていると解釈できる。したがって、これら二つの関数子カテゴリー表示は関数合成の対象となる。両者の活動的部分B(つまり、名詞句Tomの活動的部分は(S:pred tom' \backslash NP:tom')であり、他動詞eatsのそれは(S:eat' np2 np1 \backslash NP:np1)である)はユニファイされる。(8)においては、この関数合成のインスタンスエイションとリエントランスとを同時に表記する(S:eat' np2 tom' / (S:eat' np2 tom' \backslash NP:tom') / NP:np2)ことにより、多少の簡略化がなされている。続いてストリップピングによる結果として、A/Cの形式で表され

るような関数合成の結果カテゴリー表示 (S:eat' np2 tom' /NP:np2) が得られる。この関数合成の結果カテゴリー表示は、さらに次に出会う目的語名詞句applesのカテゴリー表示 (NP:apples) との間で、今度はユニフィケーションに基づく関数適用を受ける。そして、文全体の最終的なカテゴリー表示 (S:eat' apples' tom') へともたらされる。

この派生の例に見られるように、主語名詞句と他動詞とが関数合成により構成要素を形成することによって、文構造は左枝分かれ構造を示している。たとえば、次の (9) のような例は、主語名詞句と他動詞による構成要素の概念をより明確にするだろう。

(9) Harry found and I cooked the mashrooms.

このように、標準的文法理論ならば当然右方枝分かれ構造を与えるような連鎖にたいして、UCCGの関数合成は左方枝分かれ構造を与えることを許す。しかも、句が左から右へ読み進まれるにしたがって、統語情報と意味情報とが直接的な関係を保持しつつ、意味解釈が増加的になされていくことを可能にしている。

Haddock (1987) は、このように増加的に集積されていく意味表示が、聴者の実際の文処理において、そのコンテキスト内でやはり増加的に評価されていくような適切な具体例を示している。そして、そのような増加的意味解釈が統語ルールと一致するならば、文法もまた増加的であるとして、関数合成に基づく左方枝分かれ分析の有効性を指摘している。

5. 再処理機構

ところで、増加的意味解釈による左方枝分かれ派生が、与えられた句の派生において常に適切に遂行されるという訳では、勿論ない。文処理が増加的に進行していく過程で、派生がブロックされるような地点に至る場合があるからである。そのような時に、再処理機構が機能する。UCCGにおいては、そのような再処理機構は“分解”と呼ばれる手順に依っている。たとえば、次の (10) の X loves Y wildly という文の派生の例を見てみよう。

(10) X	_____ loves _____	Y	_____ madly _____
NP:x'	(S:love' np2 np1 \NP:np1) /NP:np2	NP:y'	(S:madly' s1 \NP:np1) \ (S:s1 \NP:np1)
S:pred x' / (S:pred x' \NP:x')			
S:love' np2 x' /NP:np2			
S:love' y' x' [De.]			
S:pred x' / (S:pred x' \NP:x') S:love' y' np1 /NP:y'			
S:madly (love' y' np1) \NP:np1			
S:madly' (love' y' x')			

(10) の増加的意味解釈の派生は、X loves Yにたいして S:love' y' x' というカテゴリー表示を与えた時点でブロックされる。何故なら、後続する動詞句修飾要素とは結合できないからである。そこで、分解 [De.] 手順によって既に処理した部分の再処理を行なうことになる。分解とは、 $AB \Rightarrow C$ の形式の関数適用ルールにおいて、AとCがインプットとして与えら

れたとき、Bを決定するような手順である。すなわち、この場合には、Cは見かけ上での文 $X \text{ loves } Y$ のカテゴリー表示であり、Aはそのもっとも左側の要素Xのカテゴリー表示である。したがって、Bは $\text{loves } Y$ のカテゴリー表示ということになる。こうして分解によって得られたもっとも右側の要素（すなわち、 $\text{loves } Y$ ）のカテゴリー表示は、後続の修飾要素のカテゴリー表示とユニフィケーションに基づく関数適用によって結合する。そして最終的には、その得られた要素（すなわち、 $\text{loves } Y \text{ madly}$ ）は再びユニフィケーションに基づく関数適用によって、分解によってもっとも右側に分離された要素（すなわち、X）のカテゴリー表示と結合する。こうして、分解による再処理が完了する。

勿論、分解によって分離される要素（AとB）には、いくつかの選択の可能性が存在する。どの選択をなすかは後続する修飾要素に依るだろう。いずれにせよ、分解の過程においてもっとも重要なことは、第三の見いだされるべき要素のカテゴリー表示を決定するためには、二つの要素の完全なカテゴリー表示が必要であるということである。

6. おわりに

以上の論議で、少なくともユニフィケーションの概念が、関数適用のみならず関数合成にとってもその説明原理になりうることを、したがってユニフィケーション操作が、基本的に重要な操作として増加的意味解釈を仮定するような文法の基底となりうるかもしれないことを、示すことができたと思う。

関数合成は、増加的意味解釈にとって不可欠の結合ルールであるように思われる。しかしながら、ここでは論じなかったが、この結合ルールが過剰の生成能力をもっていることはよく知られている。そこでUCCGのような関数合成を含むシステムが、そのため与えてしまうかもしれない過剰な構造を削るため、フィルターとしての制約やライセンスの問題が重要になってくるだろう。

この問題の解決法の一つは、カテゴリー表示をより豊かにすることによって得られるかもしれない。すなわち、Bach (1983) が彼の一般化カテゴリー文法において提案しているように、UCCGのカテゴリー表示に格、数、人称、その他の統語素性や意味素性などを導入することである。あるいは、さらに統語論的制約や意味論的制約を加えることも考えられる。こうして豊富なデータ構造を与えられたカテゴリー表示は、それらに加えられるユニフィケーション操作にたいする制約条件となるだろう。事実、ユニフィケーション操作は、そのような制約条件を満足させる過程である。

自然言語はローカルな、またグローバルな曖昧性に満ちている。人間の言語処理は、何らかの方法でこれらの曖昧性に対処している。増加的意味解釈は、いわば暗黙の知としてのそうした対処の方法と、決して無関係ではない。その意味で、Bresnan & Kaplan (1982) が言う能力文法 (Competence Grammar) を目指す仮定と言えるだろう。

REFERENCES

- Ades, E. and M. Steedman, 1982, 'On the Order of Words,' *Linguistic and Philosophy* 4, 517-558
- Bach, E., -1983, 'Generalized Categorical Grammars and English Auxiliary,' in F. Heny and B. Richards (eds.), *Linguistic Categories: Auxiliary and Related Puzzles*, vol. 2, Reidel, Dordrecht, 101-120.
- Bresnan, J. and R. Kaplan, 1982, 'Introduction: Grammars and Mental Representations of Language,' in J. Bresnan (ed.), *The Mental Representation of Grammatical Relations*, MIT Press, Cambridge, Mass., 1982
- Haddock, N., 1987, 'Incremental Interpretation and Combinatory Categorical Grammar,' *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, Milan, 661-663.
- Montague, R., 1973, 'The Proper Treatment of Quantification in Ordinary English,' in Hintikka et al. (eds.), *Approaches to Natural Language*, Reidel, 1973, 221-242. (Also in R. Thomason (ed.), *Formal Philosophy: Selected Papers of Richard Montague*, Yale Univ. Press, 1974, 247-270)
- Steedman, Mark, 1985, 'Dependency and Coordination in the Grammar of Dutch and English,' *Language*, 61, 523-568.
- Steedman, Mark, 1987, 'Combinatory Grammars and Parasitic Gaps,' *Natural Language and Linguistic Theory* 5, 403-439
- Steedman, Mark, 1989, 'Work in Progress: Combinators and Grammars in Natural Language Understanding,' Lecture Notes circulated at the 1989 LSA Summer Institute, University of Arizona, 1989.
- Zeevat, H. 1988, 'Combining Categorical Grammar and Unification,' in U. Reyle and C. Rohrer (eds.), *Natural Language and Linguistic Theories*, Reidel, 202-229.